PA 1090435

THE UNITED STATES OF AMERICA

TO ALL TO WHOM THESE PRESENTS SHALL COME:

UNITED STATES DEPARTMENT OF COMMERCE

United States Patent and Trademark Office

November 07, 2003

THIS IS TO CERTIFY THAT ANNEXED HERETO IS A TRUE COPY FROM THE RECORDS OF THE UNITED STATES PATENT AND TRADEMARK OFFICE OF THOSE PAPERS OF THE BELOW IDENTIFIED PATENT APPLICATION THAT MET THE REQUIREMENTS TO BE GRANTED A FILING DATE UNDER 35 USC 111.

APPLICATION NUMBER: *60/422,498*
FILING DATE: *October 31, 2002*

PRIORITY DOCUMENT
SUBMITTED OR TRANSMITTED IN
COMPLIANCE WITH
RULE 17.1(a) OR (b)

By Authority of the
COMMISSIONER OF PATENTS AND TRADEMARKS

P. SWAIN
Certifying Officer

11129 U.S. PTO

10/31/02

PTO/SB/16 (6-95)
Approved for use through 04/11/98 OMB 0651-0037
Patent and Trademark Office, U S DEPARTMENT OF COMMERCE

PROVISIONAL APPLICATION COVER SHEET

A/PROV

This is a request for filing a PROVISIONAL APPLICATION under 37 CFR 1.53 (c).

Docket Number	4147-13	Type a plus sign (+) inside this box→	+

INVENTOR(S)/APPLICANT(S)

LAST NAME	FIRST NAME	MIDDLE INITIAL	RESIDENCE (CITY AND EITHER STATE OR FOREIGN COUNTRY)
SMEETS SELANDER NERBRANT	Bernard Göran Per-Olof		Dalby, Sweden Stockholm, Sweden Osterskär, Sweden

J1040 U.S. PTO
60/422498

TITLE OF THE INVENTION (280 characters)

SECURE INSTALLATION AND UTILIZATION OF DEVICE DATA

CORRESPONDENCE ADDRESS

H. Warren Burnam, Jr.
NIXON & VANDERHYE P C
1100 North Glebe Road
8th Floor
Arlington

STATE	Virginia	ZIP CODE	22201	COUNTRY	U.S.A.

ENCLOSED APPLICATION PARTS *(check all that apply)*

☒	Specification	Number of Pages	30	☐	Applicant claims "small entity" status ☐ "Small entity" statement attached.
☒	Drawing(s)	Number of Sheets	6	☐	Other (specify)

METHOD OF PAYMENT *(check one)*

☒	A check or money order is enclosed to cover the Provisional filing fees ($160 00)/($80.00)	PROVISIONAL FILING FEE AMOUNT ($) 160 00
☐	The commissioner is hereby authorized to charge filing fees and credit Deposit Account Number 14-1140	

The invention was made by an agency of the United States Government or under a contact with an agency of the United States Government.

☒ No.

☐ Yes, the name of the U.S. Government agency and the Government contract number are:

Respectfully submitted,
SIGNATURE [signature] DATE October 31, 2002

TYPED or PRINTED NAME H. Warren Burnam, Jr. REGISTRATION NO. *(if appropriate)* 29,366

☐ Additional inventors are being named on separately numbered sheets attached hereto.

PROVISIONAL APPLICATION FILING ONLY

Burden Hour Statement: This form is estimated to take 2 hours to complete. Time will vary depending upon the needs of the individual case. Any comments on the amount of time you are required to complete this form should be sent to the Office of Assistance Quality and Enhancement Division Patent and Trademark Office, Washington, DC 20231, and to the Office of Information and Regulatory Affairs, Office of Management and Budget (Project 0651-0037), Washington, DC 20503. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS SEND TO Assistant Commissioner for Patents, Washington, DC 20231.

679861

Our Ref.: 4147-13
PE17639US00/AHE/IO

U.S. PATENT APPLICATION

Inventor(s): Bernard SMEETS
Göran SELANDER
Per-Olof NERBRANT

Invention: Secure Installation and Utilization of Device Data

NIXON & VANDERHYE P.C.
ATTORNEYS AT LAW
1100 NORTH GLEBE ROAD, 8TH FLOOR
ARLINGTON, VIRGINIA 22201-4714
(703) 816-4000
Facsimile (703) 816-4100

SPECIFICATION

SECURE INSTALLATION AND UTILIZATION OF DEVICE-SPECIFIC SECURITY DATA

TECHNICAL FIELD OF THE INVENTION

The present invention generally relates to security issues for communication over unsecure networks, and more particularly to secure and efficient procedures for providing network devices with device-specific security data, as well as general management of such security data.

BACKGROUND OF THE INVENTION

With the rapid development of Internet, packet data telecommunications networks and other communications networks, it has become increasingly more important to be able to protect messages exchanged between nodes and/or devices in the network. For simplicity, a network entity that participates in a communication will be referred to as "device". In general, this type of communication ultimately base its security on some sort of security data, e.g. a cryptographic key or secret, which is used to establish authenticity of data origin, confidentiality of message, integrity protection of message or other security services.

To facilitate the understanding of a rationale behind the present invention, it may be helpful to think of the manufacturing process of network devices, such as mobile telephones, personal computers, security gateways, firewalls, radio base stations and so forth, in large volumes. In particular, it may for example be useful to consider a device manufacturer, with limited trust in any third party (in particular third party chip manufacturers), that needs to produce devices containing tamper-resistantly protected and per-device unique cryptographic keys to a low cost.

There are several difficulties in securely and cost efficiently manufacturing devices with security data that can later be used in securing communication over insecure networks:

- To install device specific security data, different for each device. This may require entirely new manufacturing processes for some device components and thus become costly and/or inefficient.

- To place the security data in a location within the device such that it cannot be compromised or manipulated by unauthorized parties.

- To ensure that the security data is protected from unauthorized parties during the entire manufacturing process of the device. In particular if untrusted parties are involved during manufacturing, additional security management may be necessary.

- To securely manage information, related to the security data, that is needed for an authorized party to later be able to set up a secure connection with the device. E.g. if the device security data is a shared secret key in a cryptographic protocol, such as an authentication and/or encryption protocol, the same key must be available, and only available, for the authorized communications partner(s) that should be able to set up the secure connection with the device.

For example, many communication systems of today, including mobile communication systems, paging systems, as well as wireless and wireline data networks, employ authentication and encryption procedures for the purpose of improving system security and robustness. The problem of establishing secure and robust communication is encountered in many technical applications, ranging from general network communication to more specific applications such as Digital Rights Management (DRM).

DRM, for example, is a technology for protecting a content provider/owner's assets in a digital content distribution system. The technology is in most cases implemented by encrypting the content, and associating to this content a so-called license that includes the encryption key (normally in encrypted form), and usage rights describing what is allowed to do with the content.

In the equipment that will be used for rendering the content, a DRM module/agent is implemented to ensure that the rendering follows what is prescribed by the usage rights. This agent is typically implemented as a software and/or hardware module, enforcing the usage policy as stated in the license. The DRM module/agent constitutes the trusted party within the user equipment, from the point of view of the content provider. Note that the user is not a trusted party, since the user may want to circumvent the content protection and use the content without the restrictions prescribed in the license.

The problem of securing the content is partly to manage the confidentiality of the content and the integrity of the license during transport from the content distributor to the device where the content will be used. A possible realization of that is for the content provider/distributor to securely deliver to the DRM module/agent in the rendering equipment a "key encryption key", which can be used to derive the content encryption key and check the license integrity. To protect the key encryption key, device security data, unavailable to the user, could be used by the DRM module/agent. Also some information related to this security data is needed by the trusted content provider/distributor to secure the transfer to this particular device. For example, if the security data is a decryption key, the corresponding encryption key is normally needed by the content distributor/provider.

In general, there are two solutions for storing security data in a device, either on a chip or Integrated Circuit (IC) or in some sort of programmable memory, e.g. a PROM, keeping in mind that data stored on an IC is generally more protected.

Storing secret data, e.g. a device specific random number, on an IC is possible today with standard IC production tools. However, the logistics for securely passing the random number or some related data from the IC manufacturer to the device manufacturer where the IC is used is with the present techniques either infeasible/expensive and/or requires special security management for handling this data. In general the device manufacturer and the IC manufacturer may be different parties. If some security data is managed by the IC manufacturer, then this may be a security weakness, a possible target for attacks and may also increases the costs of the IC.

The same argument applies to the IC manufacturer generating and/or storing cryptographic keys on an IC on behalf of a device manufacturer.

The device manufacturer can let the IC manufacturer store, on the IC, data that is not possible to extract after IC manufacturing, unless very advanced reverse engineering is involved. However, using this device data in a security context with the help of state-of-the-art techniques requires security management in and between IC manufacturer and device manufacturer, and is either not secure or unfeasible/expensive in an industrialization process, in particular for a mass market.

The device manufacturer can insert security data into PROM thus avoiding to include the IC manufacturer as a trusted third party, and also avoiding costly changes in the IC manufacturing process. However, secrets in PROM are not as well protected against an adversary with access (even if it is just temporary) to the device. In addition the ASIC technology required for realizing PROM functionality induces considerable extra costs on the IC, for example, through additional masks in the production process of the IC.

In addition the IC manufacturer may want to limit the use of its ICs to those device manufacturers that he/she trusts or has a business agreement with.

A different but related problem is for a third party, with trust relations to the device manufacturer and/or the user, to securely communicate with the device or with a user of the device. The security management of the device-specific security data may thus require to include other parties as well.

SUMMARY OF THE INVENTION

The present invention overcomes these and other drawbacks of the prior art arrangements.

Briefly, the invention concerns an electronic circuit such as an integrated circuit (IC), which in addition to a stored secret also comprises functionality for generating or otherwise handling device-specific security data based on the stored secret. Such an electronic circuit is arranged in a network device, and the device-specific security data handled by the circuit in operation within the network device can then be used for various security operations in network communication.

More specifically, at manufacturing, a random secret is securely stored within the IC. This could be implemented in such a way that not even the IC manufacturer knows the secret. This secret data may be any arbitrary or randomly generated number. Furthermore, the IC is preferably provided with security or cryptographic algorithm(s) implemented for execution on the IC with the secret as (at least partial) input. Once installed for operation in a network device, the stored secret of the IC may be used together with the security algorithm for generation of result data. The result data computed in this way is thus specific for the particular IC and can be used as or in order to produce device-specific security data in the device, in which the IC resides.

The generated device-specific security data may subsequently be used, e.g. for securing communication over insecure networks, including Internet and cellular communication networks. Thus, the stored secret and the algorithm(s) allow

generation of security data, e.g. encryption and decryption keys, bind keys, symmetric keys, private and associated public keys and other device specific security data, which is used in the network device during operation for security services such as e.g. establishing authenticity of data origin, confidentiality of messages, integrity protection of messages.

In particular, it is clearly advantageous to be able to generate device-specific security data and provide full security functionality based on whatever secret, random data that is originally stored in the integrated circuit by the IC manufacturer. Furthermore, the IC allows generation and management of device-specific security data for a wide range of network devices, in which the IC may be arranged. In addition, since the secret data is securely stored in the IC, there is no need for any extensive security management in the manufacturing of the network device or in the distribution of integrated circuits between the IC manufacturer and the device manufacturer.

The algorithm implemented on the IC is preferably a cryptographic function or algorithm designed so that it is computationally infeasible to deduce the result of the algorithm without knowing the secret, and/or to deduce the secret from the result.

The secret may be the sole input to the IC implemented algorithm(s). Alternatively, additional input data may be supplied and used together with the secret in the algorithm to generate the device-specific security data, including cryptographic keys. The additional input data may be publicly known information, since only the owner of the device comprising the particular IC is able to generate the result due to the stored secret involved. However, in some application it may be advantageous to protect the additional input data, e.g. by means of a user-selected password.

The device-specific security data generated by means of the stored secret (and other optional input) and the algorithm may be made unavailable on the external IC programming interface and used (only) by the algorithm within the IC to perform

security operations. As a particular example, the secret may be used in conjunction with algorithm within the IC and output clear text information without revealing the secret or the device-specific security data itself.

In an especially advantageous embodiment of the invention, the electronic circuit, e.g. an IC, includes functionality for performing cryptographic computation involving the stored secret and a first input number to produce a result representation, and to output the result representation over an external circuit interface. The circuit preferably also includes functionality for performing further cryptographic computation based on the stored number and a second input number, recreating the first input number only when the second input number corresponds to the result representation. Finally, the circuit has functionality for performing an operation, preferably a security operation or security-related operation, on a third input number based on the recreated first input number.

The IC may further be provided with an authentication protocol for requiring authentication of a particular entity (network device) to grant access to a certain function, thereby restricting the usage to authorized parties. For example, only an authorized party may use the secret stored in the IC for generation of device-specific security data. Alternatively, the security data generated by the algorithm(s) in the IC may be externally available only when authorization is granted, whereas the security data is unavailable on the external IC programming interface when authorization is denied.

The invention also relates to additional security management of the device-specific security data, e.g. certification and trust delegation, in order to enable trusted third parties to communicate securely with the network device and/or user.

The invention offers the following advantages:

- Provides secure and cost-efficient implementation of device-specific security data that can be used for protecting communications over insecure channels;

- Provides protection of device-specific security data within an IC or by secret data within the IC;

- Requires only a very limited trust in the IC manufacturer;

- No security management is needed between IC manufacturer and device manufacturer;

- Secret data in the IC may be kept unavailable for any party and only negligible information of the secret need to leak during normal operation;

- Provision of device-specific security data in combination with the generic trust delegation protocol or a device certification structure provides a feasibly implementable solution to the problem of key management for secure digital rights management; and

- Generation of device-specific security data can be combined with authorization for providing only authorized access to the device-specific secret data.

Other advantages offered by the present invention will be appreciated upon reading of the below description of the embodiments of the invention.

BRIEF DESCRIPTION OF THE DRAWINGS

The invention, together with further objects and advantages thereof, will be best understood by reference to the following description taken together with the accompanying drawings, in which:

Fig. 1 is a block diagram of an embodiment of an integrated circuit implemented with a cryptographic one-way function;

Fig. 2 is a block diagram of an embodiment of an integrated circuit implemented with encryption and corresponding decryption;

Fig. 3 is a block diagram of an embodiment of an integrated circuit implemented with encryption and decryption internally linked to a security operation to prevent presentation of the device key on an external interface;

Fig. 4 is a block diagram of an embodiment of the integrated circuit of Fig. 3 with further security enhancements using an additional input key;

Fig. 5 is a block diagram of an embodiment of an integrated circuit implemented for generation of private and public keys;

Fig. 6 is a block diagram of an embodiment of an integrated circuit implemented for Diffie-Hellman shared key generation;

Fig. 7 is a block diagram of embodiment of an integrated circuit implemented for generation of private and public keys and provided with an encryption algorithm for cryptographically protecting the output private key;

Fig. 8A is a block diagram of an embodiment of an integrated circuit implemented with an authentication protocol and a Device Access Code manager for authorizing access to a stored random secret;

Fig. 8B is a block diagram of the embodiment of the integrated circuit of Fig. 8A illustrated in operation;

Fig. 9 is a block diagram of an embodiment of the integrated circuit of Fig. 1 further provided with an authorization protocol illustrated in operation when external access to generated security data is granted;

Fig. 10 is a block diagram of an embodiment of an integrated circuit such as that of Fig. 9 illustrated in operation when external access to generated security data is denied and only internal access to the security data is granted;

Fig. 11 is a block diagram of an embodiment of an integrated circuit adapted for generation of a chain of bind keys; and

Fig. 12 is a block diagram of another embodiment of an integrated circuit having an iterative implementation for generation of a chain of bind keys.

DETAILED DESCRIPTION OF EMBODIMENTS OF THE INVENTION

In the following, the invention will mainly be described with a particular exemplary scenario in mind, namely manufacturing of network devices (also called entities), including management of device-specific security data as well as usage of security data in such network devices. It should though be understood that invention is not limited thereto, and that the manufacturing scenario merely serves as a basis for a better understanding of the concepts of the invention.

As indicated in the background section, management of security data is a very important task, since the security of the entire communication relies on it. The type of management that is appropriate depends on the particular threats or attacks that the system is required to be resistant against and also what parties in the system that to some extent are trusted.

Accordingly, the parties authorized with device-specific security data may be different for different instances of the described problem. It is assumed throughout the examples that the device manufacturer is trusted with the device-specific security data, though the invention is actually not limited to that assumption. As indicated above, the chip manufacturer does not need to be trusted with the security data, though some sort of trust relation is normally assumed, e.g. that the chip manufacturer implements what is a agreed on and introduces no secret "back-doors" and so forth. It is also common that the device owner is considered a trusted party, since it usually is in his/her interest to ensure that message transfer is secure. However, this is not necessarily true and will not be assumed – a particular exemption scenario is that of DRM, as previously indicated.

The general idea of the invention for efficient generation and management of device-specific security data without extensive security management in manufacturing and distribution is to provide network devices with an integrated circuit (IC) functionality for generating or otherwise handling the device-specific security data. At the manufacturing of the IC, a random secret data is securely stored within the IC. This secret data may be any arbitrary or randomly generated integer. Furthermore, the IC is preferably provided with security or cryptographic algorithm(s) implemented for generating, either directly or indirectly, by means of further cryptographic algorithm(s), the device-specific security data with the stored secret data as input, or part of the input.

Thus, the invention provides an efficient solution for generating and/or handling device-specific security data for any network device. In addition, since the secret data is securely stored in the IC, there is no need for any extensive security management in the manufacturing of the network device or in the distribution of integrated circuits between the IC manufacturer and the device manufacturer.

The algorithm implemented on the IC is preferably a cryptographic function or algorithm designed so that it is computationally infeasible to deduce the result of the algorithm without knowing the secret, and/or to deduce the secret from the result.

The secret data may be the sole input to the IC implemented cryptographic algorithm. Alternatively, additional input data may be supplied and used together with the secret in the algorithm to generate the device-specific security data. The additional input data may be publicly known information, since only the owner of the network device comprising the particular IC is able to generate the result due to the stored secret involved. However, in some application it may be advantageous to protect the additional input data, e.g. by means of a user-selected password.

The device-specific security data generated by means of the stored secret data and the algorithm may be unavailable on the IC programming interface and used (only) by the algorithm within the IC to perform security operations.

The IC may further be provided with an authentication protocol for requiring authentication of a particular entity (network device) to grant access to certain function, thereby restricting the usage to authorized parties. For example, only an authorized party may use the random secret stored in the IC for generation of device-specific security data. Alternatively, the device-specific security data generated or handled by the algorithm(s) in the IC may be externally available only when authorization is granted, whereas the device-specific security data is kept unavailable on the external IC programming interface when authorization is denied. In the latter case, although external

access to the security data is denied, it may still be beneficial to allow internal access to the security data.

As will be described later on, the invention is well adapted for additional security management of the device-specific security data, e.g. certification and trust delegation, in order to enable trusted third parties to communicate securely with the network device and/or user.

In the following, the invention will be described with reference to a number of different illustrative examples.

Deriving security data with cryptographic one-way function

The IC manufacturer produces an IC containing a random secret C unknown to everyone as indicated above. Also in the IC is implemented a particular algorithm f, an approximation of a cryptographic one-way function using the secret C as a partial input, e.g. a Message Authentication Code (MAC) such as HMAC [HMAC], using C as the key.

During manufacturing, the device manufacturer inserts the IC received from the IC manufacturer into a particular device. By using the implemented algorithm, device specific security data can be obtained: In a controlled environment, the device manufacturer enters some data R as input to the algorithm implemented in the IC and records the result f(C, R) = B, as indicated in Fig. 1. For example, R may be a random bit string and/or some unique device identity and so forth.

This pair (R, B) can be used later, e.g. after the device has been sold to a user, by the device manufacturer or a trusted third party to communicate securely with the device. B is used to secure the communication, e.g. as a cryptographic key in a symmetric encryption algorithm or in a message authentication code. R is needed by the device to recreate B.

R can either be stored in the device during manufacturing or supplied prior to establishment of the secure communication. R does not necessarily need to be kept confidential since only with access to the right IC the relevant output B can be produced. However R may preferably be integrity protected, e.g. with B or by some out-of-band mechanism, to protect from e.g. disturbances in communication, manipulation and/or denial-of-service attacks.

An example of a particular application could be a company owning/managing a number of physically secure network nodes communicating over an unsecure network. For example, the nodes/devices could be radio base stations in a mobile network, electricity consumption metering devices, automatic drink/food resales machines. During manufacturing, the nodes are provided with IC circuits with the general structure of Fig. 1. During configuration of the nodes by the trusted staff of the company, a number of node-specific keys B are generated by the manufacturer in response to one or more input numbers R. During use, R is distributed (preferably integrity protected) to the network nodes (or stored therein during manufacturing/configuration), and input to the corresponding IC circuits to generate the node-specific keys B. Once the secret keys are shared between involved nodes, secure communication can be established by means of any conventional cryptographic protocol using B.

The physical security was assumed here, because otherwise an adversary could have access to the key(s) B, which would allow eavesdropping, spoofing of messages and other attacks. This scenario is most favorably applicable to other embodiments outlined in the following, with much less requirements on physical protection of the nodes/devices.

Optionally, if acceptable by the trust model, B may be password protected by the user (and R erased) after being calculated for the first time by the user.

Multiple pairs (R, B) may be generated and/or multiple secrets C may be implemented e.g. to enable revocation of certain security data or to differentiate between communications parties.

Note that the particular value of C is not relevant as long as it remains unknown to unauthorized parties, in particular if no one has access to C. Very limited security management is required between IC manufacturer and device manufacturer. It suffices that C is sufficiently random over a sufficiently large set and impossible to link to the particular IC. Since it is not necessary to record or derive information from C during IC manufacturing, this can be implemented within a controlled environment at the IC manufacturer.

Bind keys

In a particular example, the pair (R, B) may constitute a bind identity-bind key pair. An example of delegation of trust involving generation of bind identity-bind key pairs is specified in the so-called Generic Trust Delegation [GTD] protocol. It may be useful to review the basics of the GTD protocol. The mechanism for establishment and delegation of trust in the GTD protocol is based on the assumption that two parties A, generally a device manufacturer, and B, generally an associated device, share a (symmetric) secret. The protocol takes advantage of the fact that the device manufacturer A normally has assigned a secret device key to the device B, which device key is properly protected in the device. A third party C, having a trust relation with A, wants to communicate securely with B. As a main component, the GTD protocol includes a basic request-reply protocol, in which C requests, from A, a bind key for secure communication with B. The party A generates a bind identity, unique for the pair B and C. Then, party A derives a bind key based on the bind identity and the secret that A share with B, preferably by using a cryptographic one-way function. The bind key, normally together with the bind identity, is sent securely from A to C (the security is based on keys derived from the existing trust relation between A and C). Since B knows the shared secret between A and B, the party B can also calculate

the same bind key given the above bind identity. The latter is not confidential and may be sent to B from A or C. Accordingly, B and C can then communicate securely using the bind key. Naturally, instead of the device key itself, another key derived therefrom could be used on both sides for calculating the bind key. In this procedure, A thus "delegates trust" to C in form of the bind key between B and C.

The device manufactures never have to reveal the device key (or more generally the entity key) to any other party, since there is no need to transfer the device key outside of the device and device manufacturer. In addition, the GTD protocol does not require a single third party trusted by all device manufacturers.

The device or entity secrets never have to leave the manufacturer's domain, except in the protected area in the device where the (IC) manufacturer stored the secret during manufacturing. The manufacturer has more possibilities (and all incentives) to keep the device key or entity key secret, compared to the prior art.

The US Provisional Application 60/411,784, also referred to as the [GTD] reference, describing the GTD protocol is incorporated herein by reference.

Deriving security data with one-way trapdoor function

In the example of Fig. 1, the device manufacturer was not generally able to choose the device specific security data but had to accept whatever came out of the one-way function f. In certain applications the security data is required to have a particular format. For example, in asymmetric cryptography such as RSA or elliptic curves, the keys are not just random strings and moreover have to be chosen with caution, more of which will be described later on. In this example we present another embodiment where any particular "device key" can be chosen for a particular device.

With reference to Fig. 2, the IC manufacturer has implemented the random secret C as before and also a practical realization of a trapdoor one-way function, in this case

represented as an encryption algorithm E using the secret C as encryption key. The corresponding trapdoor inverse algorithm D, in this case performing the decryption, may also be implemented.

During manufacturing, the device manufacturer, generates any desired device-specific security data K, e.g. a cryptographic key, and inputs this to the IC performing encryption. The result E(C, K) = X is recorded by the device manufacturer and optionally stored in the device. The thus generated pairs (X, K) can be used later by the device manufacturer or a trusted third party to communicate securely with the device, in similarity to (R, B) in the previous example (note that the role of input and output is reversed).

Optionally, if the trust model so admits, X may be protected by the user, so that authentication of a user, such as entering a password of PIN must be carried out to be able to extract the security data. As before, the method is naturally extended to multiple pairs (X, K) and/or multiple secrets C. Again, the value of C is not relevant as long as it remains unknown.

A particular example of how to achieve the security management between IC manufacturer and device manufacturer is to implement into the IC public key encryption (e.g. RSA encryption) of the secret C using the public key of device manufacturer, where the public key is stored in the IC. Thus the encrypted output will only be possible to decrypt by the device manufacturer (using the corresponding private key). C will then be known to device manufacturer.

Protection of security data within the IC

The device-specific security data obtained from the algorithm implemented in the IC in the previous examples may be sensitive information used to set up secure communication channels. Depending on the trust model, an adversary may have

physical access to the device and therefore the device-specific security data should be carefully protected.

In another embodiment of the present invention, the device-specific security data may be confined to the IC and used together with implemented algorithms to perform security operations, such as e.g. decryption of data or integrity protection checks.

In the example of Fig. 2, if the encrypted key X is available, an adversary with access to the device (e.g. if X is stored) can easily get hold of the device key K. In order to protect the cryptographic key, an option is to never display them outside the IC but use them within the IC for whatever security operations that are required.

In the following example, we therefore have two layers of operations: The device key K is encrypted/decrypted with algorithms E/D, and the result is used as key in a decryption algorithm D'. The decryption algorithm D' can be substituted with another security operation, such as data origin authentication, message integrity protection and so forth, or a combination of such security operations, as is clear to anyone familiar with the field. Optionally, the operation D' could incorporate non-cryptographic functionality that is sensitive with respect to the trust model, e.g. management of data that should be available only for authorized parties and therefore remain within the IC. DRM lends a particular example to this where the decrypted high quality clear text content (such as text, audio and video) may be required to remain confidential, though a lower resolution copy is allowed to reach the rendering device. Thus D' could include decrease of resolution, D/A conversion steps and so on, and, optionally, this may be controlled based on information relating to K.

The chip received from the IC manufacturer may thus have the following functions implemented, as illustrated in Fig. 3.

During manufacturing, the device manufacturer uses the algorithm E and key C to encrypt security data K. In this case, the decryption algorithm D with key C is not accessible outside the IC. Accordingly, X can be stored, for example in PROM in the device and at the same time K may resist an attack from an adversary with access to X and to the IC programming interface. Decryption of data (or other security operation) received from device manufacturer or third party is available by entering X and the received data, *cip*, into the relevant IC interface to obtain the clear text *cle*. The security data K is thus never exposed outside the IC. In a preferred embodiment, K is a cryptographic key, as will be exemplified later on.

Optionally, if the trust model so admits, X may be password protected by the user. As mentioned in the other examples, the method is naturally extended to several instances of K and/or multiple secrets C.

In other words, the block diagram of Fig. 3 illustrates an electronic circuit for cryptographic computations that includes:

- means for permanently storing a number C not accessible over an external circuit interface;
- means for receiving first, second and third input numbers;
- means for performing first cryptographic computation E involving the stored number C and the received first input number K producing a result representation X of the first input number;
- means for output of said representation X over an external circuit interface;
- means for performing second cryptographic computation D based on the stored number C and the second input number X (input), recreating the first input number K only when the second input number X (input) corresponds to the result representation X of the first input number; and
- means for performing an operation D' on the third input number based on the recreated first input number K.

The electronic circuit is preferably a tamper-resistant circuit, such as an encapsulated integrated circuit, and the result of the operation D', preferably a security operation or a security-related or trust-related operation, is generally output over an external circuit interface. The first input number may be a cryptographic key such as a symmetric or asymmetric key. The first and/or second cryptographic computation involving the stored number may be controlled in dependence on input of a device access code into the electronic circuit, for example by allowing access to the stored input number for the above computations only when a correct device access code is entered, as will be explained in more detail later on.

In order to further enhance the security of the IC of Fig. 3, an additional input key may be employed as is illustrated in Fig. 4. Similarly to Fig. 3, during manufacturing, the device manufacturer uses the algorithm E_1 and key C to encrypt security data K_1. The obtained encrypted output X_1 may be stored in the device and subsequently be input to the associated decryption algorithm D_1. Additional security data K_2 could also be generated. This security data K_2 is preferably encrypted and provided to the device for use as input to the IC. K_2 may be generated by the device manufacturer in connection to encryption of K_1. Alternatively, K_2 may be generated by third party, e.g. a content provider or distributor, which wants to securely distribute digital data to the device. In such a case, the content provider represents K_2 as X_2 in such a way that access to K_1 is necessary for reproducing K_2, e.g. if K_1 is a private key then X_2 is the corresponding public key encryption of the key K_2. The private key could be a private key of the device manufacturer and does not have to be known by the user. The public key could be available, e.g. from a Certificate Authority of a Public Key Infrastructure. The content provider then distributes X_2 to the device. An associated decryption algorithm D_2 is implemented in the IC for decrypting the received encrypted input X_2 by means of K_1. Decryption of data (or other security operation) received from device manufacturer or a third party, e.g. the content provider, is available by entering X_1 and X_2 and the received data, *cip*, into the relevant IC interface to obtain the clear text *cle*.

Using unknown secret as or in order to generate private key

In some cases, a random number unknown to everyone can be used as or to generate a private cryptographic key. In this section we consider the exemplary case of cryptography based on discrete logarithms. As an example we will use the discrete logarithm problem over the multiplicative group of integers modulo a large prime P with generator G. An integer chosen at random from 1, ..., P-2 can be used as a private key. As illustrated in Fig. 5, we will designate this number A, which may be identical to the unknown chip secret number C or derived from the chip secret together with optional input. As before, the number A is hidden within the IC and should not be possible to extract, nor any (except negligible) information of A.

Z is a general function for generating key A based at least on the secret C. P could optionally be input to the IC which then have to generate a suitable A. Also G could be input, but the IC should then preferably checks if G is a generator of the group. A nonce generated by the device manufacturer may also optionally be input to the IC for use in the generation (Z) of the key A.

A corresponding public key P_A should be able to output from the IC, this could e.g. be G^A mod P and/or other information such as G or P. Y is a general function for generating this public key P_A, preferably based on P, G and A. The public key should be distributed in an authenticated manner to the communications partner to be able to use securely, more of which will be described later. The IC can perform public key operations D' with the private key A, such as e.g. encryption or digital signature. Specific examples are ElGamal encryption and ElGamal signature.

The secret, C, is easily generated and stored in the IC during IC manufacturing, and with the new functionality shown in Fig. 5, it is thus possible to generate an asymmetric key pair that can be used by the device in which the IC is arranged for secure communication.

Another usage of this public-private key pair is Diffie-Hellman shared key generation, as schematically illustrated in Fig. 6. The device public key $P_A = G^A \bmod P$ is exchanged for the communications partner public key $P_B = G^B \bmod P$, where B is the corresponding private key. P_B is fed into the IC and the shared secret $G^{AB} \bmod P$ is calculated. An optional random nonce may be also used in an algorithm together with the shared secret to guarantee freshness and restrict the leaking of information of the private keys. The result is a shared secret key K_{AB}, which is not externally available. The established key can then be used for a security or security-related operation D'.

More generally, if A is a private key with corresponding public key P_A in an asymmetric cryptographic scheme, with A protected within an IC, the invention also covers the case that a cryptographic key K, encrypted by the public key P_A, is decrypted and used within the IC, and not exposed outside the IC, in analogy to the previous examples.

Depending on usage, the private key may be used as a device key. Optionally, the corresponding public key may be certified by the device manufacturer, as will be exemplified later on.

In an alternative embodiment, the user generates a private key, not necessarily directly derived from the chip secret, as illustrated in Fig. 7. For example, a pseudo-random generator (PRG) using the chip secret as seed could be iterated a number of times, possibly with some additional input, and the output used as private key. As in previous examples, the private key may be hidden within the IC and the corresponding public key available outside.

Optionally, an additional nonce may be inserted by the user during generation of the key. Optionally, a Personal Identification Number (PIN) or a password mapped to a number may be the nonce or part of the nonce to enable user authentication in the sense that the PIN or password is necessary to produce the private key inside the IC.

Yet another option that can be used in conjunction with the methods above is to encrypt the private key, generated as in one of the cases above, with encryption algorithm E and chip secret C' and output the encrypted private key X. When the private key needs to be used, X is inserted at a special interface and then decrypted with D within the IC, where the private key can be used in algorithm D' as above. Optionally, X may be password protected or require other user authentication.

Authorizing the use of IC capabilities

The above examples give the device manufacturer access to the ICs capabilities at the very moment he is in possession of the IC. It might be in the IC manufacturer's interest to enforce that the device manufacturer can only utilize the IC when so being authorized by the IC manufacturer. Also or alternatively, depending on the trust model, the device manufacturer can desire to authorize which parties (if any) that should have access to the IC capabilities. This can be achieved by conditioning certain operations within the IC, based on an authentication process. Such operations could be, c.g. access to the value C for certain algorithms, output of certain values, possibly also including C, from the IC. The authentication process could be a simple maintenance/user password, but preferably involves a secure authentication mechanism such as the Fiat-Shamir protocol [FS] or other zero-knowledge protocol.

Authentication for access

We now give an example of an authentication procedure for restricting IC capabilities. In Fig. 8A, the IC, based on a public key PK, authenticates the device manufacturer secret key SK by means of an authentication protocol such as the Fiat-Shamir protocol. The authentication protocol is preferably implemented in the IC. Additionally, a challenge R is entered into the IC and a response S is returned by the IC. The pair (R, S) constitutes a Device Access Code (DAC). For example, R may be a random number, contain information of the device identity or be a hash value of such information.

In Fig. 8B, the DAC is entered to enable the use of the value C as input to an algorithm, e.g. as in one of the previous examples. The MAC circuit takes the R value of the entered DAC and computes the expected value S'. When S'=S then the value of C can be accessed by certain algorithms or certain switches may be closed. If S' is not equal to S, the other algorithms will not receive the correct value of C (or will alternatively be disabled), the corresponding switches will instead be open (or vice versa).

Note that PK has to be entered into the IC during the manufacturing. The device manufacturer typically produces key pairs and provides the IC manufacturer with a public key PK or a list of public keys. This is public information and requires no additional security management. Also, an erroneous PK would be detected during the creation of the DAC. Given the appropriate trust model, the device manufacturer may give/license the DAC to a trusted third party. The DAC may also be used to "re-program" the device, for example replacing compromised security data with new.

The method of providing conditional access to IC capabilities on authentication can be applied to any of the previous examples. We now add some more examples.

Protection of device-specific security data

In particular, the conditional access to IC capabilities can be applied to authenticate the device manufacturer during the production of device-specific security data B, such as a "bind key".

Referring back to Fig. 1, the output number B was generated based on the secret C in response to an input number R. To protect the security data B, an option is to only display it outside the IC if the proper DAC is entered into the IC, and otherwise only allow the use of B within the IC for a particular operation.

During manufacturing of the device, the security data B is derived from R, and displayed given that the correct DAC is entered, as illustrated in Fig. 9. If the correct DAC is not entered, B is only available on appropriate internal interfaces. Thus the decryption D' (or other security operation) can be performed but B is not revealed outside the IC, as illustrated in Fig. 10.

Hierarchy of bind keys

The GTD protocol can also be iteratively applied, resulting in a chain of shared secrets. The basic GTD protocol starts with two parties sharing a secret and ends with two parties sharing another secret. The procedure could be repeated iteratively, involving a fourth party that will, after the second application of the protocol, have a shared secret with one of the previous parties, and so on for higher order iterates.

It has been recognized that also the iterated GTD protocol could be implemented entirely within an IC, as illustrated in Fig. 11. The same implementation can be used on a chain of bind keys $B_1, \ldots, B_k$ with corresponding bind identities $R_1, \ldots, R_k$. They are related by $B_i = f(B_{i-1}, R_i)$ for $i=1, \ldots, k$, where $B_0 = C$.

The first bind key B_1 is deduced by the device manufacturer during manufacturing, by entering the correct DAC access code. The bind key(s) are unavailable outside the IC unless the correct access code is entered.

By supplying a sequence of bind identities, the device can subsequently calculate the corresponding bind keys and finally perform a security operation. The bind keys are unavailable outside the IC, and can not be transferred over an external IC interface by a third party that does not know the device access code. With this implementation an attacker, with physical access to the device, will only be able to decrypt a given encrypted messages but not get access to the bind keys.

Thus we have established, without any security management between IC manufacturer and device manufacturer, a set of device specific keys (B_i, i=1, ..., k), which is only available within an IC.

In the realization of Fig. 11, the bind identities R_1, ..., R_k are inserted "in parallel". Alternatively, the bind keys may be generated by an "iterative" implementation, as schematically illustrated in Fig. 12. In the example of Fig. 12, the bind identities R_1, ..., R_k, together with a number k indicating the number of required iterations, are inserted "in serial", e.g. concatenated onto an IC input interface. A built-in algorithm of the IC then iterates the function f as many times as indicated by the inserted number k, successively processing the relevant inputs ($B_i=f(B_{i-1}, R_i)$ for i=1 , ..., k and where $B_0 = C$) to output B_k to operation D'. With this modification, any intermediate bind key can be generated for protected usage with D'.

Managing security data to include trusted third party

In the previous examples we have shown embodiments of the invention resulting in the establishment of device specific security data which required very limited security management between IC manufacturer and device manufacturer. We now discuss how to handle security management if a trusted third party wants to communicate securely with the device with or without a user being involved/trusted.

The user being involved/trusted is a common scenario and needs no further explanation. In the DRM setting the user is not trusted as we described in the background section. In other settings there may not be a user during normal operation e.g. if the device runs stand-alone. In all cases the third party must access some information to be able to ensure secure communication with the intended device. This information may e.g. be a symmetric key to a device vouched for by a trusted and authorized party or a device-manufacturer signed device public key certificate used to authenticate a communication entity. We outline these two examples in more detail below.

Symmetric key delegation to third party

Consider the example of Fig. 1. As a particular instance, (R, B) could be a "bind identity" - "bind key" pair, simply referred to as a "bind pair", as in the basic GTD protocol. Thus, one or several bind pairs are generated during device manufacturing and stored by the device manufacturer. By an out-of-band arrangement, a trusted third party is in a secure manner delegated one or several bind pairs of this particular device and can then communicate securely with the device, by referring/supplying the bind identities.

By using the example of Figs. 8A-B, a bind key can be also protected against attacks from an adversary with physical access to the device, by confining the key to the IC.

The iterated GTD protocol could be achieved analogously to allow a trusted delegate to further delegate trust to parties that can communicate securely with the device. Also, the iterated bind keys can be protected within an IC, as illustrated in Fig. 11.

Alternatively, a chosen symmetric key K can be used as described in connection with Fig. 2 and Fig. 3, and the pair (X, K) can be used in the same way as (R, B) above to allow trusted third parties to set up a secure channel to a device.

Public Key Infrastructure

Consider the structure exemplified in Fig. 3. Assume that K is an asymmetric cryptographic key, e.g. a private decryption key. The following operations could be carried out in a particular secure location at the device manufacturer during manufacturing:

A private device decryption key K may be generated together with a public encryption key certificate signed by the device manufacturer's private signature key. The latter key also has a corresponding public key certificate signed by a trusted party, such as a Certification Authority of a Public Key Infrastructure (PKI), and available for a

relevant party to access, see [HAC]. The key K is fed into the IC to produce the corresponding X, which may be stored in the device. Subsequently, the private key K may be completely erased to prevent unauthorized usage. The public encryption key certificate may be placed in a publicly available certificate repository. Anyone with access to the public key can later perform encryption of data pertaining to this device. The private decryption key only exists for a short moment in the IC.

The situation is completely analogous for digital signatures, replacing "decryption" with "signature", and "encryption" with "verification" in the paragraph above, as is known by anyone familiar with the subject.

A similar, but even simpler, procedure applies to the realizations described in connection with Figs. 4-6. There, a private key is already available or generated within the IC and the corresponding public key revealed outside the IC. Thus, the device manufacturer or the user can certify/request certification of this public key and then a third party may use the certificate to enable the desired security operations.

The embodiments described above are merely given as examples, and it should be understood that the present invention is not limited thereto. Further modifications, changes and improvements that retain the basic underlying principles disclosed and claimed herein are within the scope of the invention.

REFERENCES

[FS] Fiat-Shamir identification protocol. US Patent 4,748,668.

[GTD] Generic Trust Delegation. US Provisional Patent Application 60/411,784, filed September 19, 2002.

[HAC] Menezes, van Oorschot, and Vanstone: "Handbook of Applied Cryptography", CRC Press.

[HMAC] IETF: HMAC, Keyed-Hashing for Message Authentication (RFC 2104).

ABSTRACT OF THE DISCLOSURE

Briefly, the invention concerns an electronic circuit such as an integrated circuit (IC), which in addition to a stored secret also comprises functionality for generating or otherwise handling device-specific security data based on the stored secret. Such an electronic circuit is arranged in a network device, and the device-specific security data generated by the circuit in operation within the network device can then be used for various security operations in network communication. More specifically, at manufacturing, random secret data is securely stored within the IC. This secret data may be any arbitrary or randomly generated number. Furthermore, the IC is preferably provided with security or cryptographic algorithm(s) implemented for execution on the IC with the secret as (at least partial) input. Once installed for operation in a network device, the secret within the IC can be used together with the security algorithm for generation of result data. The result data computed in this way is thus specific for the particular IC and can be used to produce device-specific security data in the device, in which the IC resides.

(Fig. 1)

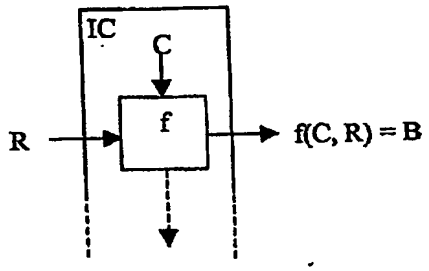


Fig. 1

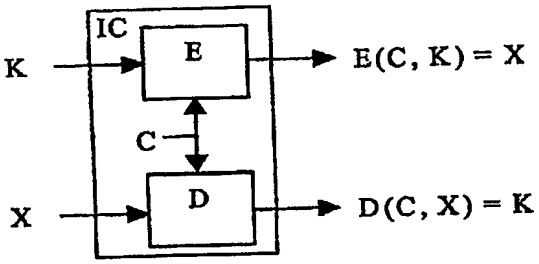


Fig. 2

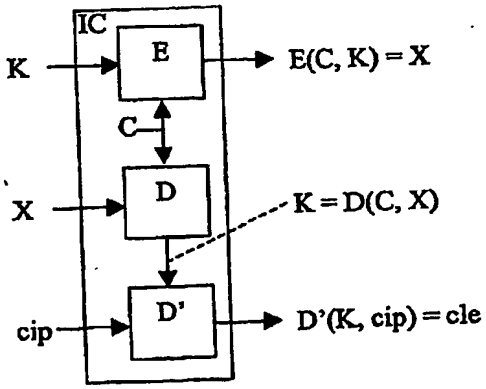


Fig. 3

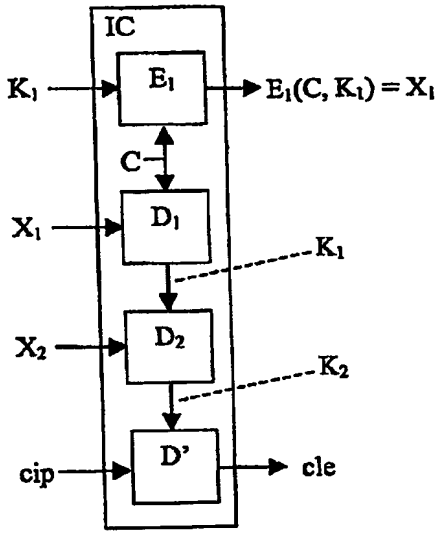

IC
K1
E1
E1(C, K1) = X1
C
X1
D1
K1
X2
D2
K2
cip
D'
cle

Fig. 4

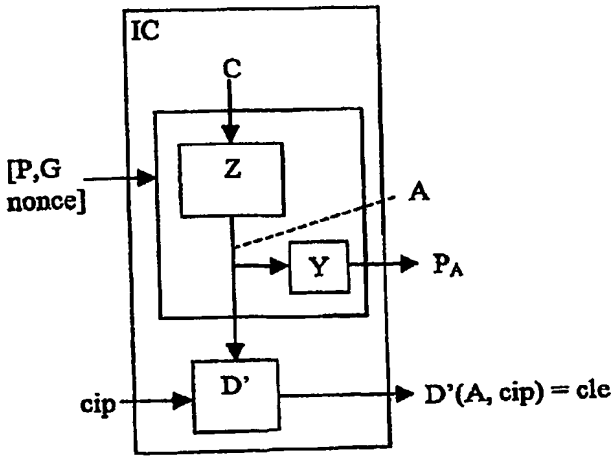

IC
C
[P,G
nonce]
Z
A
Y
PA
cip
D'
D'(A, cip) = cle

Fig. 5

Fig. 6

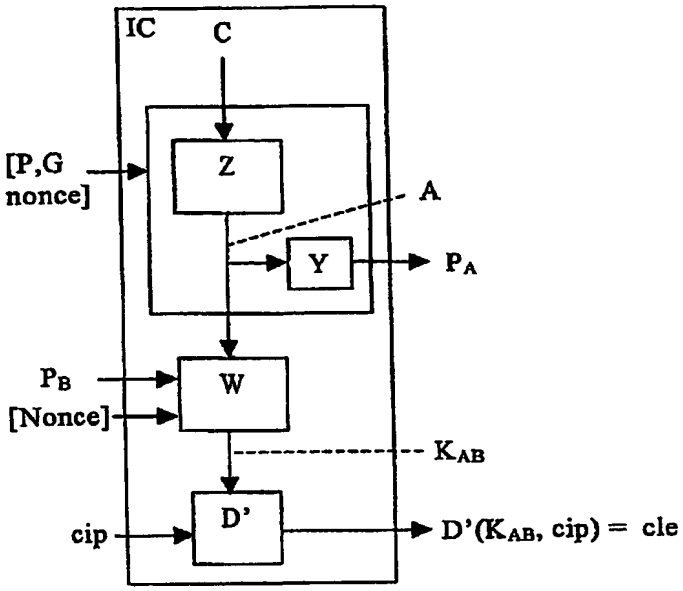
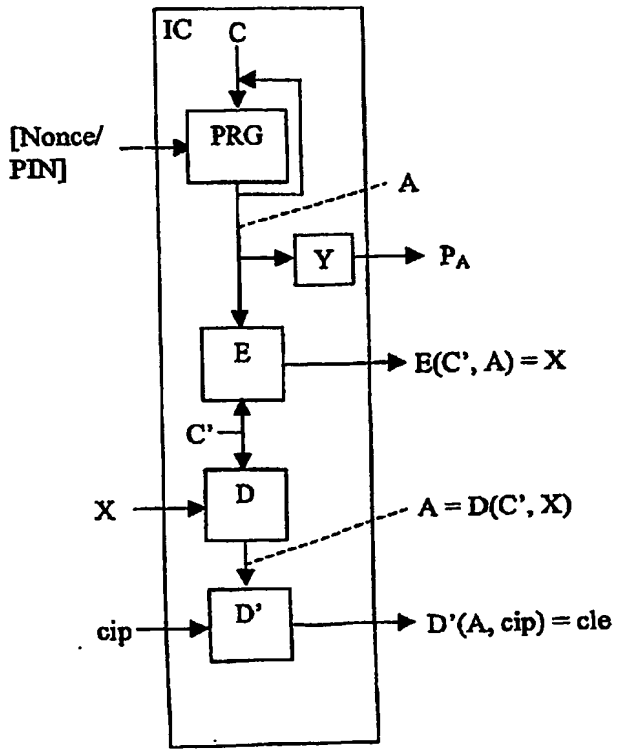

IC
C
[P,G
nonce]
Z
A
Y
P_A
P_B
[Nonce]
W
K_{AB}
cip
D'
D'(K_{AB}, cip) = cle

Fig. 7


IC
C
[Nonce/
PIN]
PRG
A
Y
P_A
E
E(C', A) = X
C'
X
D
A = D(C', X)
cip
D'
D'(A, cip) = cle

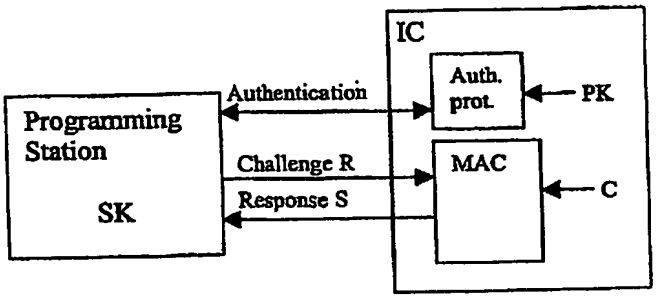


Fig. 8A

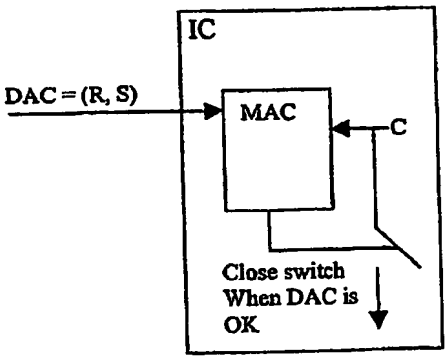


Fig. 8B

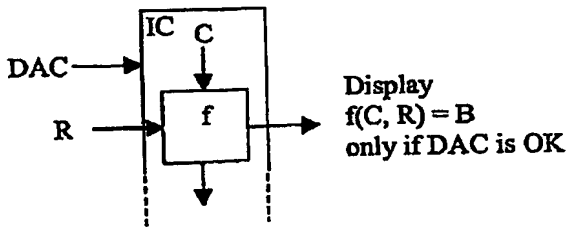


Fig. 9

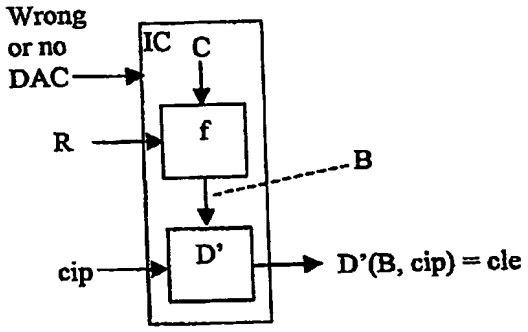

Wrong
or no
DAC
IC
C
f
R
B
D'
cip
D'(B, cip) = cle

Fig. 10

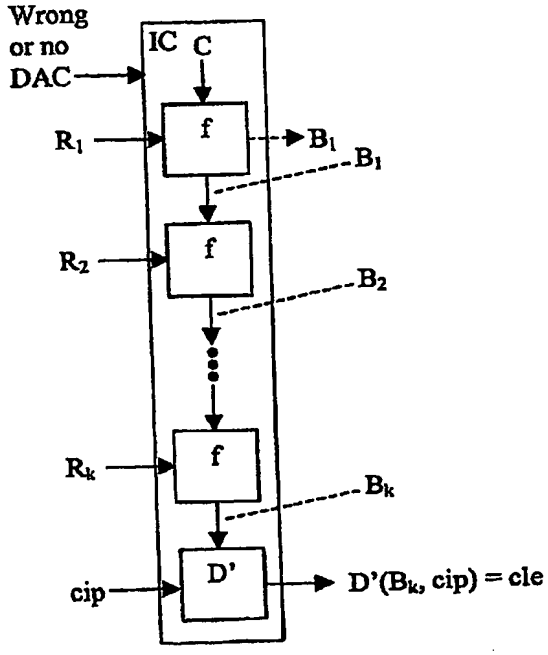

Wrong
or no
DAC
IC
C
f
R1
B1
B1
f
R2
B2
f
Rk
Bk
D'
cip
D'(Bk, cip) = cle

Fig. 11

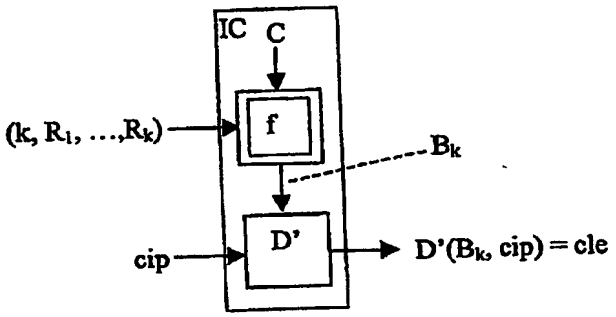


Fig. 12

This Page is Inserted by IFW Indexing and Scanning Operations and is not part of the Official Record

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images include but are not limited to the items checked:

- ☑ BLACK BORDERS
- ☐ IMAGE CUT OFF AT TOP, BOTTOM OR SIDES
- ☑ FADED TEXT OR DRAWING
- ☐ BLURRED OR ILLEGIBLE TEXT OR DRAWING
- ☐ SKEWED/SLANTED IMAGES
- ☐ COLOR OR BLACK AND WHITE PHOTOGRAPHS
- ☐ GRAY SCALE DOCUMENTS
- ☐ LINES OR MARKS ON ORIGINAL DOCUMENT
- ☑ REFERENCE(S) OR EXHIBIT(S) SUBMITTED ARE POOR QUALITY
- ☐ OTHER: _____

IMAGES ARE BEST AVAILABLE COPY.

As rescanning these documents will not correct the image problems checked, please do not report these problems to the IFW Image Problem Mailbox.